COMPONENT PART NOTICE

THIS PAPER IS A COMPONENT PART OF THE FOLLOWING COMPILATION REPORT:

TITLE: _Kalman Filter Integration of Modern Guidance and Navigation Systems._

TO ORDER THE COMPLETE COMPILATION REPORT, USE _AD-A214 284_.

THE COMPONENT PART IS PROVIDED HERE TO ALLOW USERS ACCESS TO INDIVIDUALLY
AUTHORED SECTIONS OF PROCEEDING, ANNALS, SYMPOSIA, ETC. HOWEVER, THE COMPONENT
SHOULD BE CONSIDERED WITHIN THE CONTEXT OF THE OVERALL COMPILATION REPORT AND
NOT AS A STAND-ALONE TECHNICAL REPORT.

THE FOLLOWING COMPONENT PART NUMBERS COMPRISE THE COMPILATION REPORT:

AD#: _P005 817_ thru AD#: _P005 822_

AD#: _____ AD#: _____

AD#: _____ AD#: _____

Accession For		
NTIS GRA&I		☒
DTIC TAB		☐
Unannounced		☐
Justification		
By		
Distribution/		
Availability Codes		
Dist	Avail and/or Special	
A-1		

DTIC
S ELECTE
NOV 16 1989
E D

DTIC FORM 463
MAR 85

OPI: DTIC-TID

DESIGN OF A KALMAN FILTER FOR TRANSFER ALIGNMENT

by
Dr. Oddvar Hallingstad
Norwegian Defence Research Establishment (NDRE)
P.box 25, N-2007 Kjeller
NORWAY

SUMMARY

The Norwegian Defence Research Establishment (NDRE) has been involved in the develop-
ment of several inertially based integrated navigation systems. In all of these sys-
tems, the Kalman filter has been the sensor integrator. During the last years one of
the main efforts has been on the development of the navigation system for the air
launched Penguin Mk3 missile.

Low cost inertial navigation systems (INS) are extensively applied in missile midcourse
guidance. The launch platform is generally equipped with a high quality INS, and there
is a need for some means to transfer this performance to the missile INS. This is done
by transfer alignment (TA) before launch. This alignment may in general be achieved by
angle, position, velocity or acceleration matching (alone or in combination).

This paper describes the design philosophy used in the development of the alignment
subsystem of the inertial midcourse navigation system for the air launched Penguin
antiship missile adopted for the F-16 fighter aircraft. The desired performance was
achieved through a three level Kalman filter (KF) design process. On the first level
we assume that our system is linear and then we design the KF. On the second level we
deal with the design of the preprocessor which make the linear assumptions on level one
valid. The last level deals with the field testing of the missile navigation system
which is the final test of the validity of the design procedure.

LIST OF SYMBOLS AND ABBREVIATIONS

The notation used in this article is based on reference [10].

g	Free fall acceleration	T	KF update time interval
IMU	Inertial measurement unit	$(\cdot)$	Vector or matrix belonging to the system truth model.
INS	Inertial navigation system		
MINS	Master INS (aircraft)	$(\cdot)^d$	Vector or matrix belonging to the filter design model.
SINS	Slave INS (missile)		
KF	Kalman filter	$(\cdot)^m$	Vector belonging to the MINS.
TA	Transfer alignment	$(\cdot)^s$	Vector belonging to the SINS.
x,y,z	The three body axes		
X,Y,Z	The three navigation frame axes	$(\cdot)_k$	Time k

1 INTRODUCTION

The F-16/Penguin is an anti sea invasion weapon system with a high performance missile
designed to take optimum advantage of the confined Norwegian coastal waters. To protect
the aircraft and missile and avoid missile impact on land the missile has a high navi-
gation accuracy independent of both aircraft and missile trajectories. The heart of
this system is the missile INS, a relatively low cost semi strapdown INS (the roll axis
is gimballed) based on two two-axes gyroscopes and three accelerometers with a turn-on
to turn-on accuracy of the order of deg/h and mg (milli g), respectively.

The missile has to be able to fly a variety of attack profiles in order both to avoid
and to make use of the mountainous Norwegian coastal terrain. One of several attack
sequences is illustrated in figure 1. The missile will in typical operational scen-
arios experience heavy manoeuvres, both high g turns and linear accelerations in 3
dimensions, both immediately before launch and in free flight. This puts heavy demands
on the inertial midcourse navigation system and the prelaunch initialisation procedure.
In this paper, the main topic will be the design of this prelaunch initialisation pro-
cedure.

The original transfer alignment (TA) problem was to estimate the mechanical misalign-
ment between the case axes of two IMUs. Since our IMUs have been turned into full
fledged inertial navigation systems (INSs), it turns out that the TA problem may be
formulated as an ordinary navigation system update problem. Because the MINS is order
of magnitudes more accurate than the SINS, the output from the MINS may be considered
error free. Thus the original TA problem has been transformed into the navigation prob-
lem : Estimate the velocities and the misalignments in the SINS using the velocity

outputs from the MINS as measurements of the true velocities. The Kalman filter (KF) is used to solve this problem. The reason why we use velocities instead of positions as measurements to the KF will be discussed in subsection 3.1.2.

In section 2 we will describe the transfer alignment problem in more detail and discuss the design objectives and the design procedure for a Kalman filter meeting the specifications. The design procedure consists of three levels. On the first level we assume that our system is linear and then we design the KF (section 3). On the second level we will in section 4 discuss the design of the preprocessor which make the linear assumptions on level one valid. The last level discussed in section 5, deals with field testing of the missile navigation system. This is the final test of the validity of the design procedure.

2 TRANSFER ALIGNMENT

2.1 System Description

The alignment is done by matching the outputs from the F-16 INS and the missile INS by means of a KF. The design objectives and procedure for the KF are given in the next sections. The KF transfer alignment algorithm is implemented in the software for a Motorola 68000 based microcomputer in the Penguin/F-16 adapter. The adapter fits between the standard F-16 pylon and the missile.

Figure 2 shows a physical block diagram of the main components in the TA system. The INS in the F-16 aircraft is a gimballed 3 axes platform while the INS in the missile is a semistrapdown platform (the roll axis is gimballed). The accuracy of MINS is several magnitudes better than the SINS implying that the MINS may be considered error free for TA purposes.

The accelerations and angular velocities sensed by the two platforms differ due to the spatial separation and the nonrigid body connecting them. This nonrigid body is susceptible to both mechanical deformations and vibrations.

The available navigation data on the F-16 1553B and the missile buses are updated with 50 Hz. This is obviously too much for a KF. So there must be some form of averaging of the measurements. The time lags for the data from the two sources are also different.

2.2 Design Objectives and Constraints

The most important design criteria was that the missile INS alignment should not impose heavy restrictions on the normal operation of the aircraft. That is, there should be no added restrictions on g loads imposed by the alignment subsystem, and of particular importance, the alignment should not impose restrictions on aircraft manoeuvres during the launch sequence. In addition, the alignment subsystem should not require any changes on the aircraft. Some of the design objectives and constraints for the KF are :

*   The navigation accuracy (position and attitude) should meet the specifications at the target.
*   The alignment time must be shorter than the requirements.
*   The filter should only utilize the readily available velocity and attitude data on the F-16 1553B data bus.
*   The computation load and memory requirements have to fit into the available Motorola 68000 based micro computer in the missile adapter.
*   The filter has to be robust. I.e. unexpected large or unknown error sources should not cause major performance degradation.
*   The filter may be turned on at any time and then stay on even during long missions.

2.3 Design Procedure

Designing a KF for TA meeting the design objectives given in the previous section may be done using the following three level iterative procedure :

Level 1 : Kalman filter design

On this level we assume that the system equations have been linearised so that the KF may be applied. This assumption depends on the success of the preprocessor design on level 2. The design is assisted by a covariance analyses simulation program.

1. Put the problem into a KF framework.
2. Decide whether an optimal KF may do the job (disregarding computation load and memory requirements) or not.
3. Eliminate states from the optimal KF arriving at a suboptimal filter.
4. Tune the suboptimal filter.

5.   Perform a sensitivity analysis to determine the robustness of the suboptimal
     filter. Repeat from 2 if necessary.

Level 2 : Preprosessor design

The purpose of the preprocessor is to interface the KF designed on level 1 to the
physical system. The design is made using a Monte Carlo simulation program where the
main nonlinear aspects of the physical system are implemented.

1.   Decide how to perform level arm compensation and calculation of the KF measure-
     ments.

2.   Determine the discretisation algorithm of the timevariant matrices used by the KF
     and based on output from the SINS.

3.   Determine the KF update frequency.

4.   Design a supervisor which detect abnormal situations, i.e. hardware failures, out-
     layers and abnormal signal statistics.

5.   Perform a Monte Carlo simulation incorporating the KF from level 1.

6.   Repeat from 1 if necessary. If the KF is inadequate repeat level 1.

Level 3 : Field testing

Both level 1 and level 2 designs were based on simulation programs. On level 3 the
algorithms found on the previous two levels are implemented in the alignment unit and
tested in the physical system. The test results are analysed using a post flight simu-
lation program.

1.   Implement the algorithms in the alignment unit hardware.

2.   Perform captive flight tests.

3.   Perform missile test firing

5.   Analyse the test results using the post flight simulation program or if necessary
     the Monte Carlo simulation program and the covariance simulation program.

6.   Repeat the level 1 and 2 design if necessary.

3 KALMAN FILTER DESIGN

We will in this section show how to deduce the KF part of the alignment algorithm. The
KF design is based on several assumptions which will be tested in section 4 and 5. The
design objectives and constraints given in subsection 2.2 may for the KF design in this
section be taken care of as follows :

*   If all the assumptions for a KF are valid it will be optimal, implying that the
    navigation accuracy at the target and the alignment time cannot be made better by
    any other estimation method. Therefore, test of the optimal KF will tell whether
    these requirements are achievable or not.

*   The optimal KF design will pose unacceptable computation load and memory require-
    ments. We have to design a suboptimal KF. The deduction and test of this suboptimal
    KF that preserve the optimality is therefore the main concern in this section. The
    computation load will be reduced by eliminating states, simplifying the matrix
    structure and by updating the KF with a much lower frequency than the measurement
    frequency (section 4).

3.1 System Truth Model

We will in this subsection present the system truth model and its properties. The sys-
tem truth model is the best, most complete mathematical model that can be developed.
For our KF design purpose it is linear and serves both as a starting point for the
suboptimal filter design and as a reference for achievable alignment accuracy.

3.1.1 Nonlinear Model

A block diagram of the main components in the physical TA system is shown in figure 2.
A mathematical model of the process part would consist of the following models (as our
Monte Carlo simulation program does) :

1.   A trajectory generator which calculates the linear and angular acceleration inputs
     to the MINS and SINS $(\vec{a},\vec{\omega},\vec{a},\vec{\omega})$. The generator may be designed in many different
     ways. In our Monte Carlo simulation program we first specify a trajectory (curve)
     consisting of line and circle segments in the computation frame. Then we specify
     the tangential acceleration involved. The model of the aircraft is fairly simple
     because we specify that the normal component of the acceleration is always normal
     to a plane through the wings of the aircraft. The actual linear $(\vec{a})$ and angular

$(\vec{\omega}^s)$ accelerations for the MINS may then be calculated. The input $(\vec{a}^i, \vec{\omega}^i)$ to the SINS is calculated by adding the level arm effect $(\vec{\omega} \times \vec{r})$ and the output from a vibration model to $\vec{a}^s$ and $\vec{\omega}^s$.

2. A model of the gyros and accelerometers in the aircraft gimballed inertial platform and the navigation equations implemented in the MINS. In most of our Monte Carlo simulations we were only interested in relative navigation errors. Thus, we used the positions and attitudes given in the trajectory generator directly.

3. A model of the gyros and the accelerometers in the missile semistrapdown inertial platform and the navigation equations implemented in the SINS. The SINS's navigation equations used the quaternion algorithm.

The common way to apply the KF to a nonlinear system is by using an extended Kalman filter (EKF) where one has to implement nonlinear models of the dynamics and the sensor equations in the computer. The transfer alignment problem may also be solved using an EKF. But the nonlinear dynamic equations mentioned above would be to much for a real time application. Instead, one makes a linear error model of the difference between the outputs from the MINS and SINS. This will be done in subsection 3.2. As a matter of fact also this error modelling procedure may be interpreted as making an EKF. In this case the SINS is interpreted as the dynamic nonlinear model of the true aircraft dynamic. That is, the nonlinear aircraft dynamic equations are solved on a combined analog-digital computer (the SINS) and the SINS is reset by the KF error estimates. It is this feedback which makes the filter an EKF and not a linearised KF in this interpretation.

3.1.2 Linearisation

The KF is an algorithm which is optimal only for linear gaussian systems, but most of the real world problems are nonlinear (including our TA problem). A main question is therefore how to linearise the process in figure 2 in order to make the KF algorithm applicable. We make the following assumptions :

1. The MINS is considered error free because its accuracy is several orders of magnitude better than the SINS. The navigation data from the MINS are consequently taken as true positions, velocities and angles. For filter design purposes only the relative estimation errors are of interest. The absolute navigation error may be obtained by calculating the RMS of the relative navigation error and the absolute navigation error in the F-16 INS.

2. The preprosessor (discussed in the next section) compensates exactly for the spatial separation of the two platforms by compensating for the level arm effect $(\vec{\omega} \times \vec{r})$ and averaging out any vibration differences. Thus, we assume that the two platforms are sensing the same linear and angular accelerations.

3. We assume that an initial coarse alignment has been done, making the axes misalignment so small that a linear error model is valid.

4. In order to keep the misalignments small the SINS will be reset by the KF error estimates. That is, the SINS will be in closed loop during alignment.

These assumptions render the linear TA truth model given in figure 3. The validity of the assumptions will be tested by Monte Carlo simulations and field tests. Thus, the difference between the MINS and SINS measurements may be modelled by a linear time variant stochastic model of the form (the linear truth model):

$$\underline{\dot{x}}'(t) = F'(t)\underline{x}'(t) + G'(t)\underline{w}'(t)$$

with a discrete measurement model :

$$\underline{z}'_k = H'\underline{x}'_k + \underline{v}'_k$$

Table 3-1 shows an example of a system truth model and state variables. The three position states are not included. But they are needed for evaluation purposes as the alignment has to be evaluated according to the position and level errors at the target. Consequently, the position states are included in the simulation programs but not in the implemented KF.

No. of States	Description	Abbreviation	Initial values (1-σ
3	Relative velocity error (X,Y,Z)	V	0.4 m/s
3	Relative attitude error (X,Y,Z)	E	16 mrad
3	Gyro coloured noise (x,y,z)	GYCN	2 deg/h (30 s)
3	Gyro bias (x,y,z)	GYB	6 deg/h
3	Gyro scale factor (x,y,z)	GYSF	0.20 %
4	Gyro mass unbalance (x,z)	GYMU	6 deg/h/g
3	Accelerometer bias (x,y,z)	ACB	2 mg
3	Accelerometer scale factor (x,y,z)	ACSF	0.26 %
6	Accelerometer nonorthogonality (x,y,z)	ACNO	0.4 mrad
1	Relative azimuth measurement bias	AMB	16 mrad

Table 3-1 System truth model, state variables

The initial error truth model of inertial platforms are easily set up by using the accelerometer and gyro models from the producer in addition to information of the actual mechanisation. But this initial truth model contains up to 100 state variables. We arrived at the state vector in Table 3-1 by sensitivity simulations. The initial model was excited by different trajectories and only the states showing the greatest response was kept in the system truth model in table 3-1.

The readily available measurements to the alignment filter are velocity and azimuth differences between the MINS and the SINS. The purpose of the azimuth measurement is to prevent azimuth instability during nonmanoeuvering periods. Due to deficiencies in the down channel (Z-axis) of the F-16 inertial navigation system the Z-axis velocity difference is not used for the time being. The system truth measurement model i. given in table 3-2.

No. of meas.	Type of measurement noise	Abbreviation	1-σ values
3	White velocity meas. noise (X,Y,Z)	VMN	0.4 m/s
1	White azimuth meas. noise	AMN	8 mrad

Table 3-2 System truth measurement model

Table 3-2 shows that we intend to use velocity and not position as measurement. In an INS the position is only an integration of the velocity. Position may therefore not contain more information about the errors in the SINS than the velocity does, see reference [11]. Because the computation load for a KF is proportional to n^3 where n is the no of states in the KF, we decided not to use position as measurement to the KF. Further, due to our inaccurate SINS (compared to the MINS) we assume the MINS to be error free. The position outputs from the MINS are therefore used to update the position of the SINS directly.

Using positions as measurements to the KF would have the following advantage : The errors due to unmodelled kinematical motion of the SINS relative to the MINS would be more averaged than using velocities, allowing a longer KF update interval. But the simulations show that altogether we are better off using only velocities.
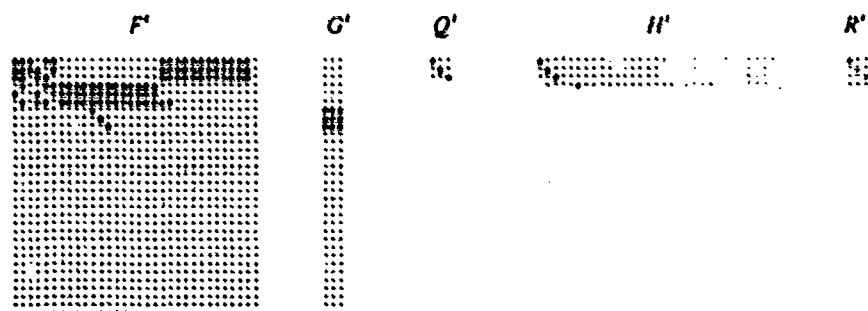

Table 3-3 Truth Model Matrix Structure

Table 3-3 shows the truth model matrix structure. A nonzero element is marked by 'x' and a zero element by '•'. The matrices are sparse due to all the bias states. The nonzero system matrix (F') elements are of three main categories :

1. Elements depending on the specific force measurements (output from the accelerometers). These elements are large and manoeuvre dependant (depending on linear accelerations).

2. Elements coupling the component errors into velocity and level error states. These elements are elements in coordinate transformation matrices from the platform gyro and accelerometer frames to the navigation frame. These elements are attitude dependant.

3. The rest of the nonzero elements (like the Coriolis coupling)

Our simulation programs utilises the structure by dividing the matrices into submatrices and eliminating multiplication with a zero submatrix.

3.1.3 Optimal KF

The alignment time and navigation accuracy at the target depends on the prelaunch aircraft manoeuvres. Fortunately, the simulations of a KF based on the system truth model show that all these requirements are fulfilled if the aircraft make only a minor prelaunch turn. This indicates that normal aircraft manoeuvres will be sufficient. The optimal prelaunch manoeuvre would be the free flight trajectory of the missile. Because then all the error sources of the SINS would have been excited also during alignment and estimated. But also a normal mission shows usually more than sufficient manoeuvers. In addition to aligning the reference axes, the alignment procedure will also to a certain extent calibrate the inertial sensors' bias and scale factors.

In order to calculate the navigation and alignment accuracy the aircraft and missile trajectories have to be defined. To simplify the problem somewhat, this paper will consider two stylistic situations of a minimum alignment and a complete alignment.

Minimum alignment time is the time required for the alignment filter to estimate pitch and roll attitude errors. This does not include warm up, power on test, and initial coarse alignment. Minimum alignment is defined as an alignment where there has been no manoeuvres to make the azimuth error observable. This information is available for the pilot as status information on the F-16 stores control panel.

Complete alignment requires that an aircraft manoeuvre has made the azimuth attitude error and other manoeuvre dependent accelerometer and gyro errors observable. In addition, the alignment time has to be long enough so that the estimates of attitude and gyro biases have stabilized.

Figure 6 shows the trajectory used in the generation of the error budget in figure 4. This trajectory is sufficient for a complete alignment.

Figure 4 shows the error budget for the position and level error states at the target. In the figure we have combined the effect from all three axes for each kind of error. This is not done in the original simulation and we are there able to distinguish between the axes. The dominating error sources for the position errors (given the trajectory in figure 6) are the gyro coloured noises (GYCN) and the velocity measurement noises (VMN). The GYCNs are nonobservable due to a 30 s correlation time. The VMNs are also dominating because we assume them to be large due to nonmodelled vibration noise.

For the level errors the dominating error sources are the GYCN, the gyro scale factor (GYSF) and the gyro mass unbalance (GYMU). The GYMU is not observable for nonmanoeuvering cases. For our trajectory we are not able to separate it fro the gyro bias.

An analysis of the error equations with respect to observability gives the following results :

*   Azimuth error is observable through the velocity measurements given a manoeuvre in the horizontal plane. The level errors are also observable without manoeuvres due to the free fall acceleration g.

*   Because the roll angle is always zero, the y-gyro coloured noise, bias and scale factor are not observable through the azimuth measurement. But they may be estimated through velocity measurements due to the level error to velocity couplings.

*   In order to estimate the accelerometer scale factors and nonorthogonality and the gyro mass unbalance the aircraft must have manoeuvres.

3.2 Filter Design Model

The optimal KF tested in the previous section satisfied the accuracy requirements at the target. But a 32 state KF would be too much for the available micro computer. Hence, in this subsection we will try to reduce the computer demand by eliminating states, simplifying the system matrices and discretisation algorithm.

3.2.1 State Elimination

The computation load of a KF may be decreased by :

* Eliminating state variables in the filter model.
* Replacing coloured noise state variables by bias states (time update of bias states is not necessary).
* Simplifying the discrete error model used for time update calculations.

The candidates for elimination are nonobservable states and states which give a small contribution to the total navigation error at the target. Nonobservable states may eventually be replaced by an observable linear combination. The effect of state elimin- ations and other simplifications should always be checked by a full covariance simula- tion. Simulations of different trajectories suggested that the following states may be eliminated from the list in table 3-1 :

* 6 accelerometer nonorthogonality states because their influence on the navigation accuracy is small for the majority of manoeuvres.
* 3 gyro coloured noise states because their observability is low (30 s correlation time) and their main effect of keeping the KF gains up may be replaced by white process noise on the velocity and angular levels. Elimination of these coloured noise states is also important because it leads to a significant reduction of KF time update computation time.
* 1 x-gyro scale factor because the SINS is roll stabilized.
* 4 gyro mass unbalances because their effect on the navigation accuracy is neglect- able (the effect is manoeuvre dependent).

The number of states in the filter model is now 18 versus 32 in the system truth model (disregarding the three position state variables which will not be implemented in the final filter). The last 12 state variables are modelled as biases which will give an insignificant contribution to the time update computation load. The total computation load is now acceptable.

The problem with the present filter model is that the KF gains for the bias states will approach zero. This may imply filter divergence due to all the unmodelled states. Fig- ure 5 shows the true alignment accuracy for the X-axis in the missile platform as calculated by the covariance simulation program. The divergence problem will be addressed in subsection 3.3.

3.2.2 Matrix Simplification

In section 3.1.2 we discussed the structure of the linear truth model. Many of the couplings shown are of minor importance in an INS like ours because of the large compo- nent errors. Therefore, we eliminate all the couplings due to Coriolis-accelerations and error in the calculation of the g-vector. These simplifications will also speed up the on line calculation of the elements in the design model matrices. The structure of the design model is shown in table 3-4. Notice the introduction of white process noise on the velocity and angular levels.

$$F^* \qquad G^* \qquad Q^* \qquad H^* \qquad R^*$$

Table 3-4 Design Model Structure

3.2.3 Discretisation

The connection between the matrices in the linear, timevariant, continues stochastic differential equation

$$\underline{\dot{x}}^*(t) = F^*(t)\underline{x}^* + G^*\underline{w}^*$$

and the discrete difference equation

$$\underline{x}^*_{i+1} = \Phi^*_i \underline{x}^*_i + \Gamma^*_i \underline{w}^*_i$$

is given by :

$$\dot{\Phi}^*(t,t_o) = F^*(t)\Phi^*(t,t_o) \qquad \Phi^*(t_o,t_o) = I$$

$$\Gamma_k^d Q_k^d (\Gamma_k^d)^T = \int_{t_k}^{t_{k+1}} \Phi^d(t_{k+1}, \tau) G^d Q^d (G^d)^T (\Phi^d(t_{k+1}, \tau))^T d\tau$$

We will comment on the calculation of these matrices for long intervals in subsection 4.2. If the interval is short enough, we may calculate the matrices by Taylor series expansion. As table 3-5 shows, the F^d-matrix is nilpotent and the Taylor series for F^d is :

$$\Phi_k^d = I + F^d T + \frac{1}{2}(F^d T)^2$$

An approximate solution of the integral for $\Gamma_k^d Q_k^d (\Gamma_k^d)^d$ is easily found by using the formula for the Φ_k^d-matrix. As table 3-5 shows the structure of these matrices is sparse.
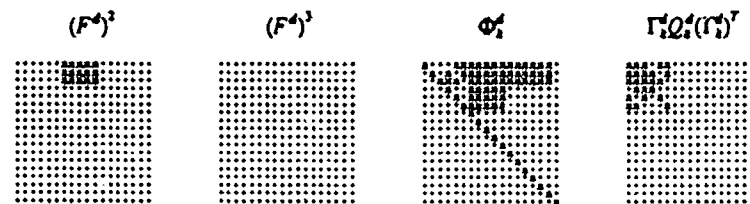

Table 3-5 Design Model Discretisation

3.3 Filter Tuning

As shown in figure 5, a simple elimination of state variables gives a divergent KF. We have therefore introduced fictitious process noise in order to keep the KF gains at an acceptable level. The computation burden is kept low by introducing white noise only to the velocity and error angle equations in the filter design model. In addition we want to make the filter robust. This is done by making the a priori covariances for the design model larger than the nominal values. The filter performance degradation due to these conservative design filter values is small for nominal values in the truth model.

Figure 5 shows the tuned alignment accuracy for the X-axis of the platform. The true estimation error is now not distinguishable from the optimal estimation error. A similar comparison of KF gains shows that the tuned filter has almost optimal gains. The sensitivity of the tuned filter model to changes in the process model was investigated by the covariance simulation program. The following cases were examined :

- Initial velocity and level errors an order of magnitude greater than nominal.

- Unexpected bias shifts in the y- and z-gyros.

- Many different alignment manoeuvres including the extremes : no manoeuvres at all and very violent manoeuvre.

- Long term stability.

The conclusions from all these simulations were that :

1. The tuned KF satisfies the accuracy requirements with only modest demands on the alignment manoeuvres

2. The suboptimal KF gives almost the same position and attitude accuracies at the target as the optimal KF. Also the important KF gains are almost identical to the optimal case. This is achieved by introducing fictitious process noise on the attitude rate error level.

3. The filter is robust due to relative large a priori covariances

4. The calculation time for the KF time and measurement updates is fast due to an UD-algorithm utilising the matrix structures.

The results of these simulations, and also evaluation through Monte Carlo simulations and experience from captive flight testing, led to the final tuned filter model in table 3-6.

No. of States	Description	Abbreviations
3	Relative velocity error (X,Y,Z)	V
3	Relative attitude error (X,Y,Z)	E
3	Gyro bias (x,y,z)	GYB
2	Gyro scale factor (x,y,z)	GYSF
3	Accelerometer bias (x,y,z)	ACB
3	Accelerometer scale factor (x,y,z)	ABSF
1	Relative azimuth measurement bias	AMB

Table 3-6 Tuned filter model, state variables

3.4 UD-factorisation algorithms

The UD-factorisation algorithm (reference 2) was used both in the simulation programs and the implemented KF. The use of numerical stable algorithms is a necessity in covariance simulation programs due to the high dimension of the augmented state vector. In the real 'me implementation the conventional covariance equations would probably have been sufficient. But since the computation burden is almost equal to the UD-algorithms the latter were chosen.

The UD-algorithms were simplified in order to utilize the special structure of the equations at hand (this kind of simplifications is much easier to do to covariance equations). Especially the elimination of the gyro coloured noise states were important in keeping the computation load small.

Notice, that the UD-algorithm has to use both Γ_i^I and Q_i^I in the update equations. This is done by Cholesky factorisation algorithm.

4 PREPROSESSOR DESIGN

The KF designed in section 3 satisfied the design requirements. But the KF was designed making several assumptions about the preprosessor function. The simulations in that section did not account for those assumptions. In this section, we will show how the preprosessor is designed and give some results of the Monte Carlo simulations. This simulation program accounts both for the most important nonlinear effects and the difference between the INS output frequencies and the KF update frequency. The preprosessor main structure is given in figure 7.

4.1 Kalman Filter Measurements

The accelerations and angular velocities sensed by the two platforms differ due to the spatial separation and the nonrigid body connecting them. The nonrigid body is susceptible to mechanical deformations and vibrations. Thus, the velocity measurements from the SINS have to be corrected for the level arm effect and the vibrations. Since we cannot update the KF with 50 Hz, we calculate the average effect of the level arm over one KF update time interval and correct the average velocity measurements from the SINS before we form the difference with the averaged velocity measurements from the MINS. This averaging leads to a reduction in the measurement noise and averages the high frequency vibrations. But it could lead to stability problems due to the new correlation between process and measurement noise. Fortunately, simulations show that this added correlation is too small to make any problem for our KF.

4.2 Time Variant Matrices

The most correct way to calculate Φ_i^I and $\Gamma_i^I Q_i^I (\Gamma_i^I)^T$ is by using a sub interval, ΔT given by the INS measurement frequency. The Φ_i^I-matrix is then given by

$$\Phi_i^I = \Phi_1(\Delta T)\Phi_2(\Delta T)...\Phi_n(\Delta T)$$

where $\Phi_i(\Delta T)$ is calculated by the formula in subsection 3.2.3 and using the most recent acceleration measurements in the calculations. Because the computation load would be too great, we first calculate the average F-matrix for the KF time update interval and then calculate Φ_i^I according to the formula given in subsection 3.2.3. Simulations verify that this is satisfactory for our system.

4.3 Kalman Filter Update Frequency

In order to check the robustness of the KF for the chosen KF update frequency, T, we run the Monte Carlo simulation program with update intervals from T/2 to 4T. The simulations showed no significant difference in the estimation errors. The chosen update frequency, measurement calculation and matrix calculation methods are therefore judged to be healthy.

4.4 Error Checks

In the application of KFs to real systems it is of vital importance to realize that abnormal situations will arise. Hence, some kind of error detection and status indication has to be build into the system. The tests to be implemented may be foreseen to a certain extent, but due to the hardware dependence, field tests have to be done. The final test limits are determined through a close interplay between simulations and field tests. This interplay will be discussed in the next section, but the actual tests will be presented here.

The KF assumes that the measurement statistics are given. Due to hardware deficiencies, outlayers which violates these statistics have to be expected. These outlayers are eliminated be using a 3-σ test on the innovation process.

The azimuth angle is observed through velocity changes in the horizontal plan. During periods with small velocity changes (manoeuvres) the azimuth angle is nonobservable through the velocity measurements, but the azimuth angle measurement maintains the accuracy by relying on the aircraft INS and assuming no relative rotation of the two body axes. The azimuth measurement is not used when the manoeuvres exceed a given limit. However, during manoeuvres the missile and aircraft axes may move relatively to each other creating a new permanent offset. This is modelled by reinitialisation of the azimuth measurement bias.

Due to hardware failures the KF may diverge. Such divergence may be detected by monitoring and checking the calculated variances in the KF and the mean and standard deviation of the innovation processes.

The component estimates (gyros and accelerometers) are checked and error flags are set if the estimates exceeds certain limits, thus indicating component failure.

The alignment accuracy depends on the alignment time and manoeuvres. The KF covariances and the mean and standard deviation of the innovation processes are used to calculate a performance index. This index tells the pilot if a manoeuvre may enhance the alignment or not.

4.5 Monte Carlo Simulation

The performance of the navigation system has been evaluated using both covariance and Monte Carlo simulations. Monte Carlo simulations involve multiple runs of a simulation including all known noise and error sources to establish accumulated statistical properties of selected state variables as a function of time. And, as opposed to a covariance analysis, a Monte Carlo simulation has no inherent restrictions to the implementation of the models involved. E g there is no need for a linearised model for the generation of the measurements to the suboptimal KF. Computer cost is the major disadvantage of the Monte Carlo validation technique.

We have done a lot of Monte Carlo (MC) simulations in order to verify the KF design from section 3 and the preprosessor design in this section. All of the simulations show close agreement with the covariance simulations. The differences are well within the statistical limits (based on 100 MC runs). In addition to covariance calculations the MC program may also calculate the mean values. Also these simulations give values within the statistical limits.

The alignment is close to optimum. It is very robust, and there is little to be gained by expanding it.

5 FIELD TEST RESULTS

5.1 Captive Flight Tests

The flights have not been planned specifically for evaluating the navigation system, but rather as rehearsals for the actual missile firing. Alignment times have varied. Most of them have represented complete alignment, and only a few have had minimum alignment (see subsection 3.1.3 for definition). 4) tests have been evaluated, non of which have been identical.

The position error has been calculated by integrating the velocity difference between the aircraft and the missile INS, with corrections for relative movements. The navigation error has been well within specifications and in close agreement with predicted performance from simulations. A few of the 43 tests were close to minimum alignments.

5.2 Missile Test Firings

A number of missiles has been launched during the engineering development and technical evaluation phases. It has been difficult to isolate the missile INS error from test range instrumentation errors, F-16 fire control errors, and F-16 INS errors, however, the missile firings indicate a close agreement with computer simulations and captive flight testing.

5.3 Deficiencies and Errors

Initially our most serious problem was telemetry dropouts. Especially it was difficult to receive reliable data during aircraft manoeuvres. The situation was gradually improved during the test period. Among improvements were better aircraft tracking equipment on the ground and merging of data from more than one telemetry receiver antenna. On the software side a lot of effort had to be put into program modifications in order to handle data dropouts and unreliable data. Heavy restrictions on the flight trajectories were imposed by telemetry coverage and general air safety restrictions in the test area. In fact, the captive flight testing was an integrated part of the normal fighter pilot training in southern Norway. Our testing should not interfere with the normal operation of this airfield. A telemetry pod on the F-16 aircraft itself would have spared us a lot of problems. This was not available at that time, however, today this is an integrated part of the test equipment.

Our next serious problem turned out to be the weather. When finally the aircraft, the missile, and the telemetry and data reduction system, all, from a technically point of view, were ready for testing; wind, ice and snow quite a few times turned out to be the final reason for the cancellation of the take off. After all, our aircraft was not on alert, and the pilot had to follow peacetime general air safety precautions. However, we had to adapt to this situation as well. Through simulations we realised that a lot could be done by just taxing on the runway. Especially, the identification of quite a few time tag errors was done by data from F-16 pirouettes on the runway. The fighter pilots did not actually love the test trajectory when a test engineer again and again ordered a 720 degrees pirouette just to have another look at his KF states.

The captive flight test period gave us, as mentioned, new knowledge about system behaviour. However, due to concurrent effort in testing, simulation, and new algorithm development it turned out to be very easy to identify these deficiencies when they appeared and to make appropriate software changes. Two examples were changes necessary for the compensation of relative motion on the azimuth measurement and changes to the use of this measurement, a result of new knowledge about how the pilot operated the aircraft, and how the missile was mounted to the aircraft, respectively. Another example is the fact that manoeuvres early in the fine alignment period with relatively low constant g loads introduce delays in the alignment. Minor changes had to be introduced in the setting of status information on the stores control panel display. And in general, as expected, several test limits and the initial uncertainties of a few filter states had to be slightly adjusted. However, our major problems were due to true errors as listed below.

The time tag and KF prefilter software, or the synchronization of missile and aircraft data, the major part programmed in assembly and fix point arithmetic, turned out to include a lot of errors. All of them had to be identified and removed to achieve a reliable alignment performance. The most difficult time tag error to identify was one which caused altitude information to be put into the least significant bits on the F-16 time tag. This error was identified and removed when we realised that the delay caused by time tag was a function of aircraft altitude.

Incorrect sign on different terms in the missile INS software was another problem. All of these errors were, except for one, identified and corrected before or early in the captive flight test period. The one left over, due to inconsistencies in the documentation, was an incorrect sign in one of the terms for compensation of the movement over the Earth. This error was in some instances equivalent to a gyro bias of more than 10 the nominal value, and the alignment filter decomposed this error as different component errors as a function of aircraft manoeuvres. We finally identified this error by code inspection when telemetry data had told us that the error was a function of velocity and heading in level flight. On reflection, this particular error, and maybe some of the time tag problems, may have been sorted out by simulations before the captive flight testing started. This was, at the time, not possible mainly because the INS navigation software, the alignment software, and the time tag software all were, with a few exceptions, tested independently by different people. A closer integration during testing may have been a wiser approach. However, of particular importance is the fact that the simulation software eventually had to be designed by the same people implementing the necessary real time system software; thus, more effort put into simulation, with the same resources for the total job, may have delayed the captive flight testing. We had to decide on a priority. We had prepared for open loop testing of the

alignment filter, testing without actually updating the INS with error estimates. However, open loop testing turned out to be of little practical use as the INS diverged too fast for the isolation of the relatively small error effects. To sum up; our experience is that it is easy, through inspection of KF behaviour, to tell if the system behaviour is different from expected. However, to isolate the error source is very difficult and a time consuming iteration between testing and detailed software code inspection in a lot of different subsystems. When the source finally is identified the necessary modifications are fast and easily included in the stage of development described here.

6 CONCLUSIONS

The alignment subsystem imposes no additional restrictions on the operation of the F-16 fighter aircraft. There is no need for a particular alignment trajectory, and, the fighter pilot may switch on the alignment at any time, e g when he is checking other subsystems on the ground before the mission. From the KF design approach a high performance, highly reliable and robust missile midcourse inertial navigation system emerged. Navigation accuracy is well within specifications. The reasons for the success were in the first place the effort put into simulations both during alignment filter development and during the captive flight testing, and secondly the extent of data collected during testing for performance analyses. If we should have done the job over again, we would have put even more effort into the simulations, the telemetry system and the post processing of telemetry data. Finally, the transfer alignment unit was possible to test and evaluate through relatively inexpensive captive flight testing.

REFERENCES

1. Gelb, A. (ed.), Applied Optimal Estimation, M.I.T. Press, Cambridge, Mass., 1974.

2. Thornton C.L. and Bierman G.J., "UDU^T Covariance Factorization for Kalman Filtering", Control and Dynamic Systems, Vol. 16, Academic Press.

3. Bar-Itzhack, I.Y. and Porat, B., "Azimuth Observability Enhancement During Inertial Navigation System In-Flight Alignment", Journal of Guidance and Control, Vol.3, July-Aug. 1980, pp. 337-344

4. Porat, P. and Bar-Itzhack, I.Y., "Effect of Acceleration Switching During In-Flight Alignment", Journal of Guidance and Control, Vol.4, July-Aug. 1981, pp. 385-389

5. Sutherland A.A. Jr. and Gelb A., "The Kalman Filter in Transfer Alignment of Airborne Inertial Guidance Systems", NWC TP 4653, Naval Weapons Centre China Lake, October 1968

6. Baziw J. and Leondes C.T., "In-Flight Alignment and Calibration of Inertial Measurement Units - Part I: General Formulation", IEEE Trans. Aerosp. Electron. Sys., Vol.AES-8, No.4, July 1972

7. Baziw J. and Leondes C.T., "In-Flight Alignment and Calibration of Inertial Measurement Units - Part II: Experimental Results", IEEE Trans. Aerosp. Electron. Sys., Vol.AES-8, No.4, July 1972

8. Kraemer J.W., Roessler J.J. and Aradin D.M., "In-Flight Alignment/Calibration Techniques for Unaided Tactical Guidance", NAECON 1978 RECORD, pp. 705-711, May 1978

9. Kortüm W., "Design and Analysis of Low-Order Filters Applied to the Alignment of Inertial Platforms", AGARD lecture series No. 82 on Practical Aspects of Kalman Filtering Implementation, March 1976

10. Maybeck P.S., "Stochastic Models, Estimation, and Control", Volume 1, Academic Press 1979

11. Farell J.L., "Transfer Alignment for Precision Pointing Applications", NAECON 1979 RECORD, pp. 1033-1039
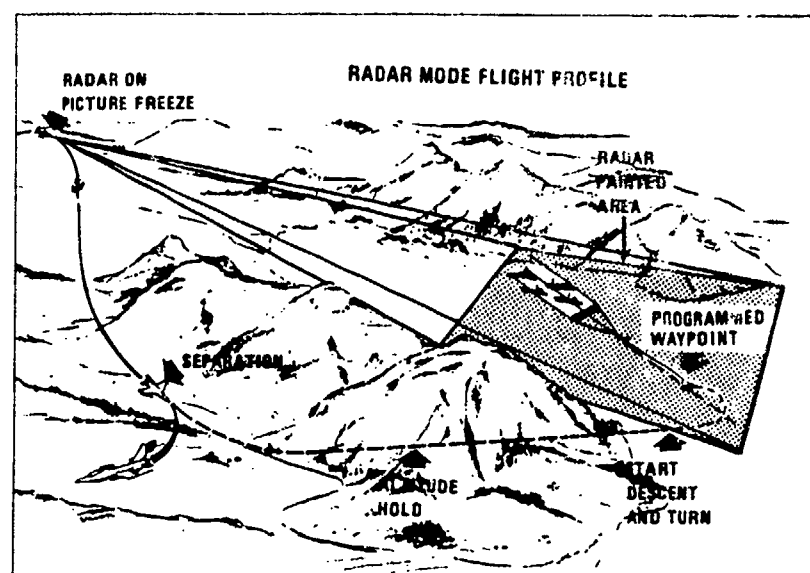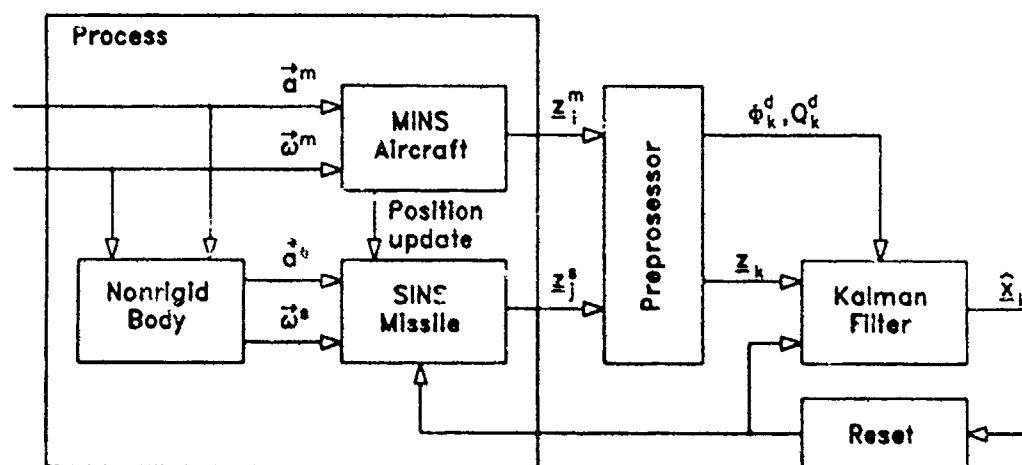
Figure 1   Radar Delivery Mode

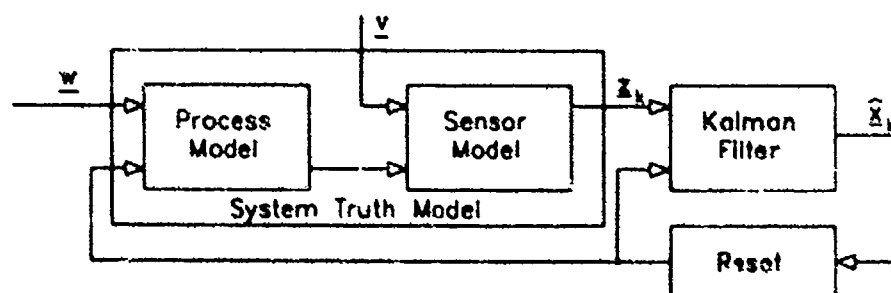Figure 2   Physical System Block Diagram
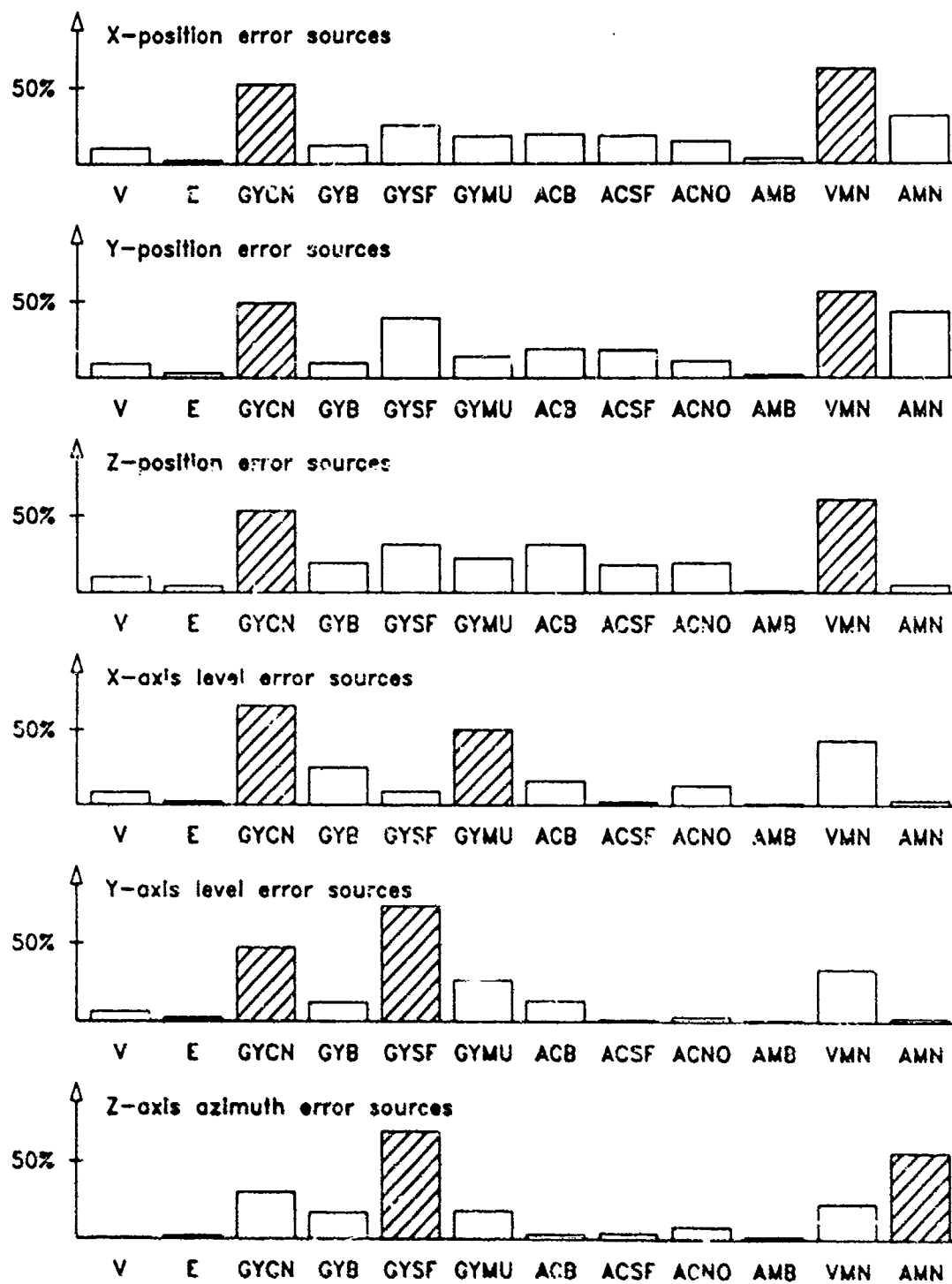

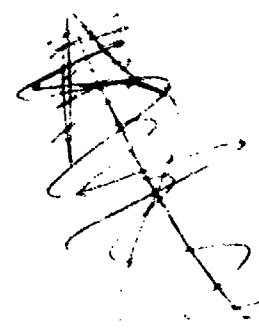Figure 3   Linear Transfer Alignment Model
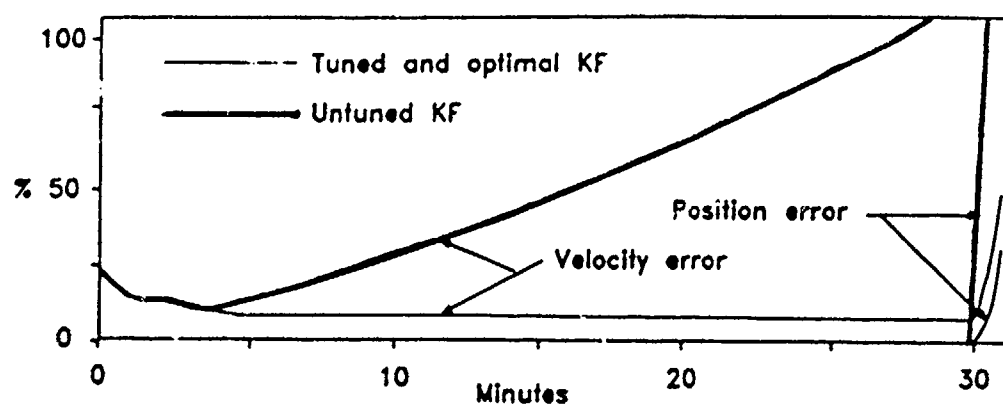
Figure 4   Optimal System Error Budget at the Target

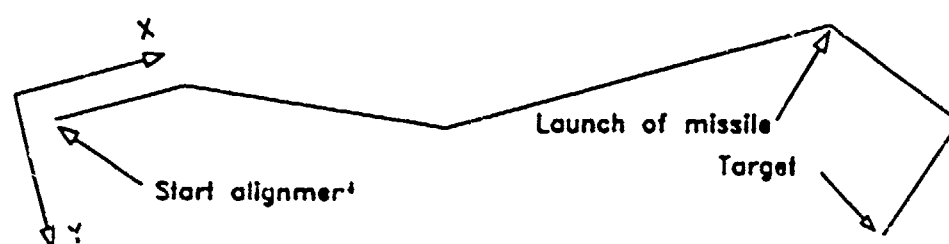Figure 5   True SD for Optimal, Untuned and Tuned KF
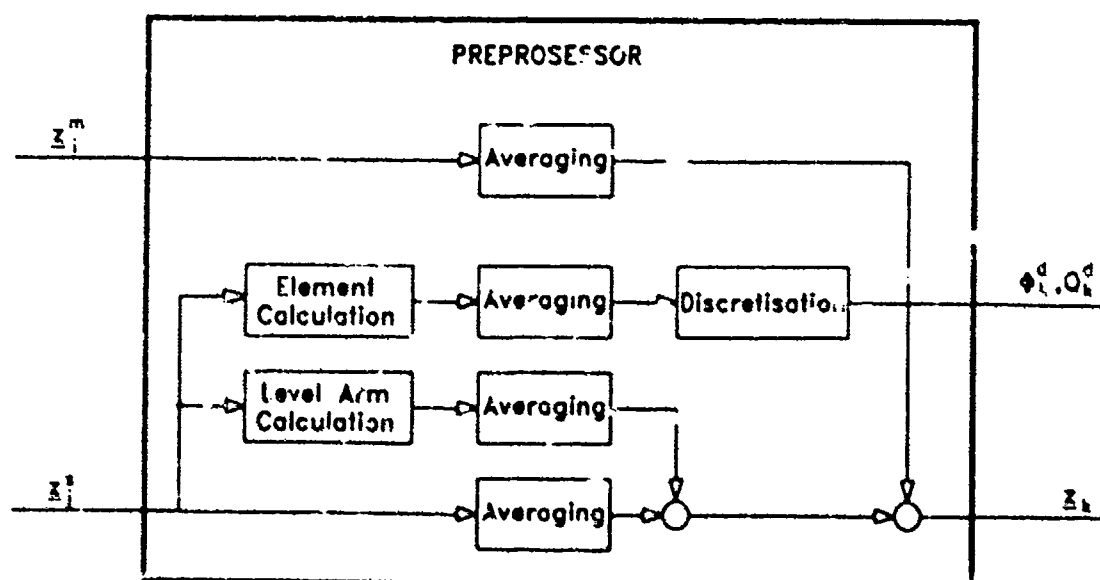

Figure 6   Sample Trajectory in the X-Y Plain

Figure 7   Preprocessor Block Diagram